Getting Started with Mixly

Guide Content

1. Introduction for Mixly 4
2. Design Concept and User Groups 6
2.1 Design Concept 6
2.2 User Groups 8
3. Interface Functions of Mixly 9
3.1 System Functions 9
3.2 In/Out Block 11
3.3 Control Block 16
3.4 Math Block 22
3.5 Text Block 27

3.6 List Block 30
3.7 Logic Block 33
3.8 Variable Block 36
3.9 SerialPort Block 39
3.10 Communicate Block 45
3.11 Sensor Block 49
3.12 Actuator Block 52
3.13 Monitor Block 55
3.14 Functions Block 59
4. Resources 64

1. Introduction for Mixly

Mixly is a free open-source graphical Arduino programming software, based on Google's Blockly graphical programming framework, and developed by Mixly Team@ BNU.
It is a free open-source graphical programming tool for creative electronic development; a complete support ecosystem for creative e-education; a stage for maker educators to realize their dreams.
Although there is an Ardublock graphical programming software launched by Arduino official, Ardublock is not perfect enough, and many common functions cannot be realized.

The figure below shows the functional comparison between Ardublock and Mixly.

	If/ else	for / while loop	Math & Boolean	Bluetooth	IR	Interrupt	Serial Communicate
Ardublock	√	√	√	√	×	×	×
Mixly	√	√	√	√	√	√	√
	Lists	**Pulse**	**ShiftOut**	**Variables Declare**		**Procedure**	
Ardublock	×	×	×	only support Integer/ Float		Not return data	
Mixly	√	√	√	support Integer/ Float /Char/ Boole		Return data	

It can be said that Mixly is the most versatile and smoothest Arduino graphical programming software, which can replace the Arduino programming tool IDE.

2. Design Concept and User Groups

2.1 Design Concept

(1) Usability

Mixly is designed to be completely green. Currently Mixly supports win, ubuntu, mac. Windows users can download the Mixly package directly from the Internet, and unzip it to run on Windows XP and above (download link is attached below).

(2) Simplicity

Mixly uses the Blockly graphical programming engine to replace complex text manipulation with graphical building blocks, providing a good foundation for beginners to get started quickly.

① Use the different color icons to represent different types of functional blocks, very convenient for users to classify.

② Provide default options in the composite function block to effectively reduce the number of user drags.

③ Integrate all the features of the software in the same interface.

④ Provide the reference tutorial and code examples.

(3) Functionality

It has versatile functions. Mixly can almost implement all the functions that Arduino IDE has. Support all official development boards of arduino.

(4) Continuity

The goal of the graphical programming system is definitely not to replace the original text programming method, but to better understand the programming principles and program thinking through graphical programming, and lay the foundation for future text programming.

It is also the design philosophy for Mixly. More continuous content has been added to the design of the software to protect the user's learning outcomes. To be specific, it includes the introduction of variable types, the consistency of text programming as much as possible in the design of the module, and the support of both graphical and text programming.

(5) Ecological

The most important design concept of Mixly is its ecological feature, which can distinguish it from other

Arduino graphical programming.
In order to achieve sustainable development, Mixly is designed to allow manufacturers to develop their own unique modules (currently supports DfRobot, StartLab, MakeBlock, Sense, Seeed, Lubot. But users require JavaScript programming foundation to make this part of the module).
It also allows users directly use Mixly's graphical programming function to generate common modules (such as LED digital display, buzzer broadcast, etc. Users are able to make this part of the module only using Mixly).
Both of the two kinds of modules mentioned above can be imported into the Mixly system through the "Import" function, thereby realizing the user's own value in the popularity of Mixly software.

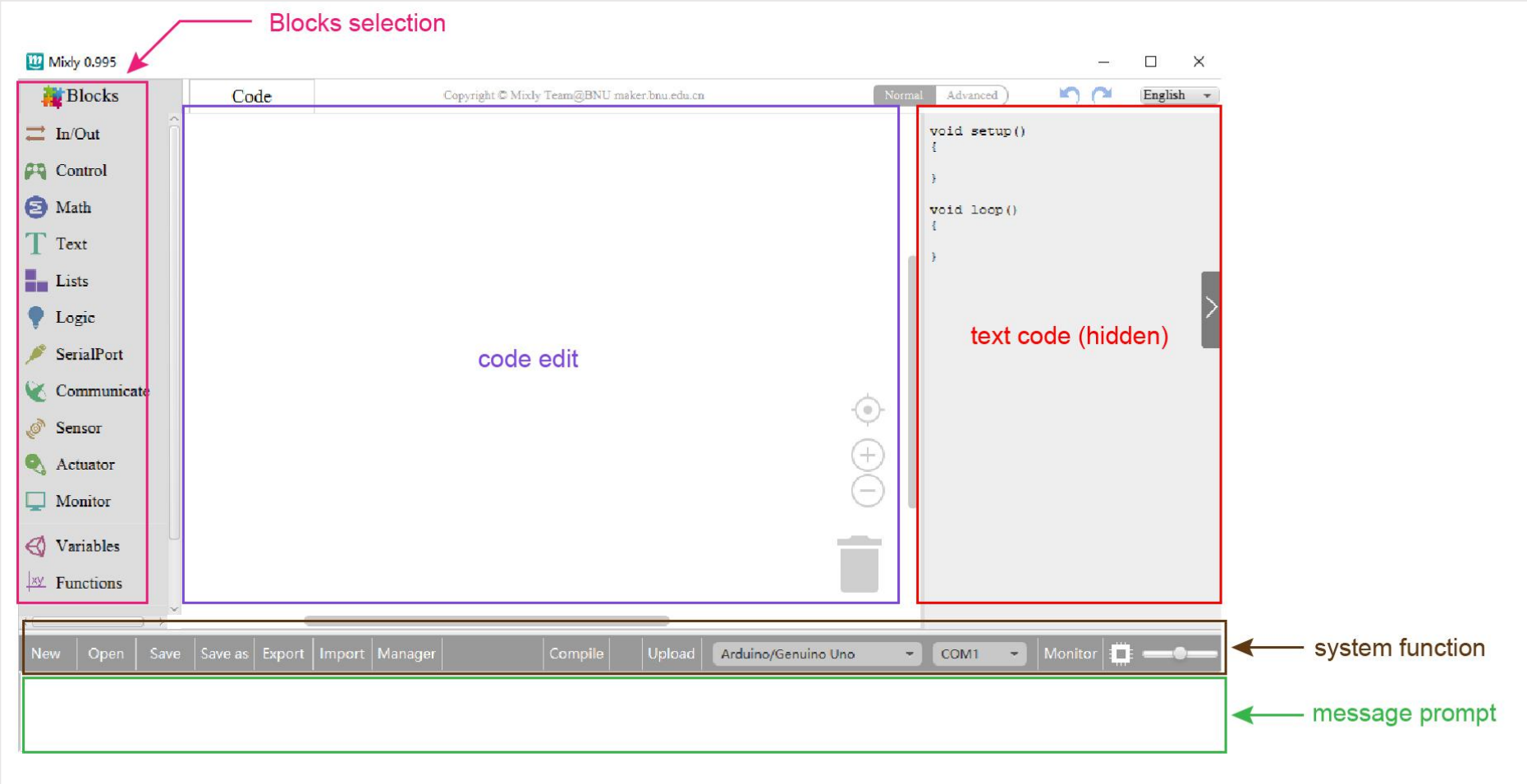
2.2 User Groups

From the above design concept, it can be seen that Mixly is suitable for primary and secondary school students to cultivate programming thinking. It is also available for quick programming when creating a work. Of course, it is good for those lovely friends who don't want to learn text programming, but want to do some small works with intelligent control.

3. Interface Functions of Mixly

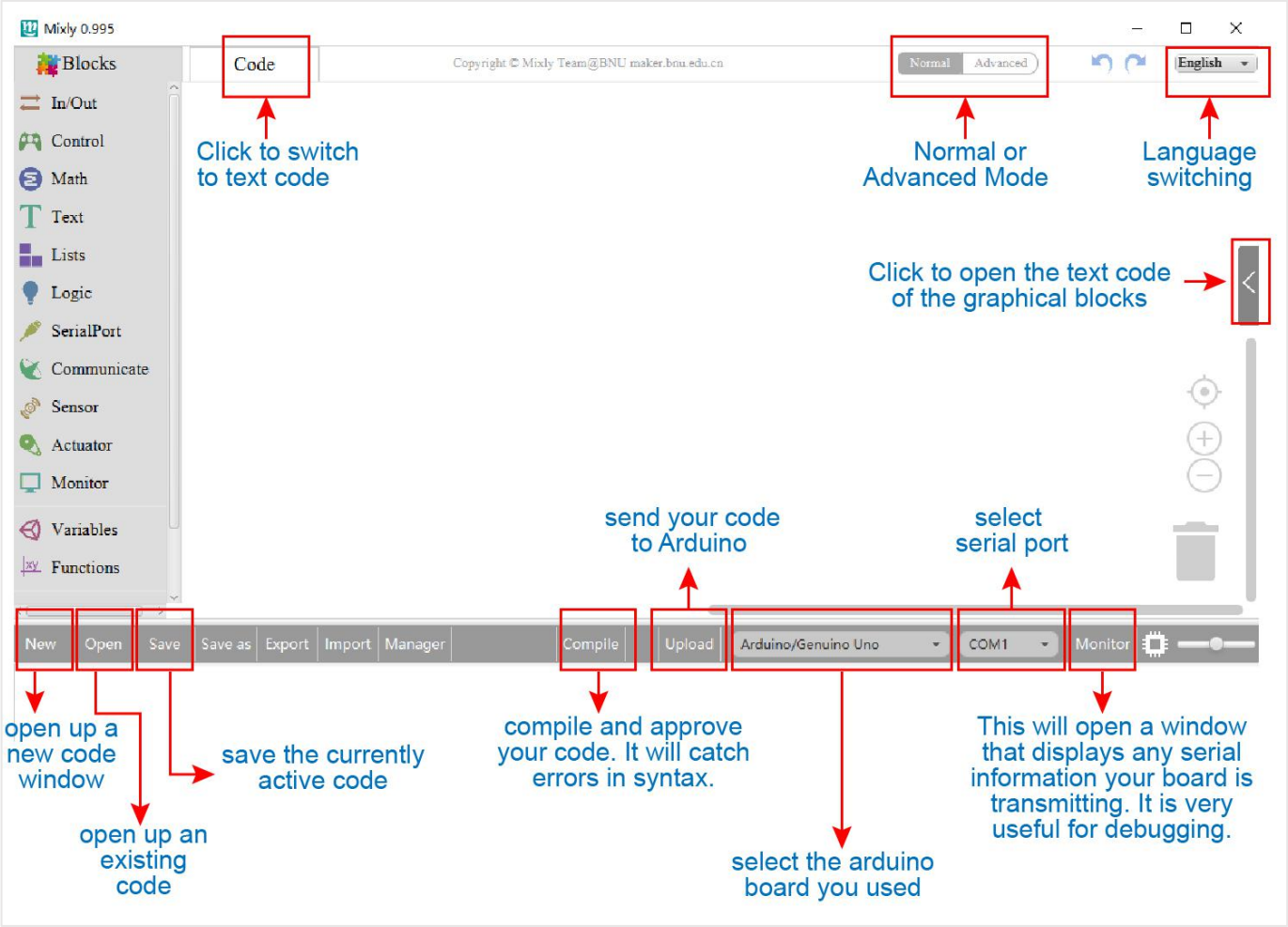
3.1 System Functions

Look at the main interface of Mixly, it includes five parts, that is, Blocks selection, code edit, text code (hidden), system function and message prompt area. Shown below.

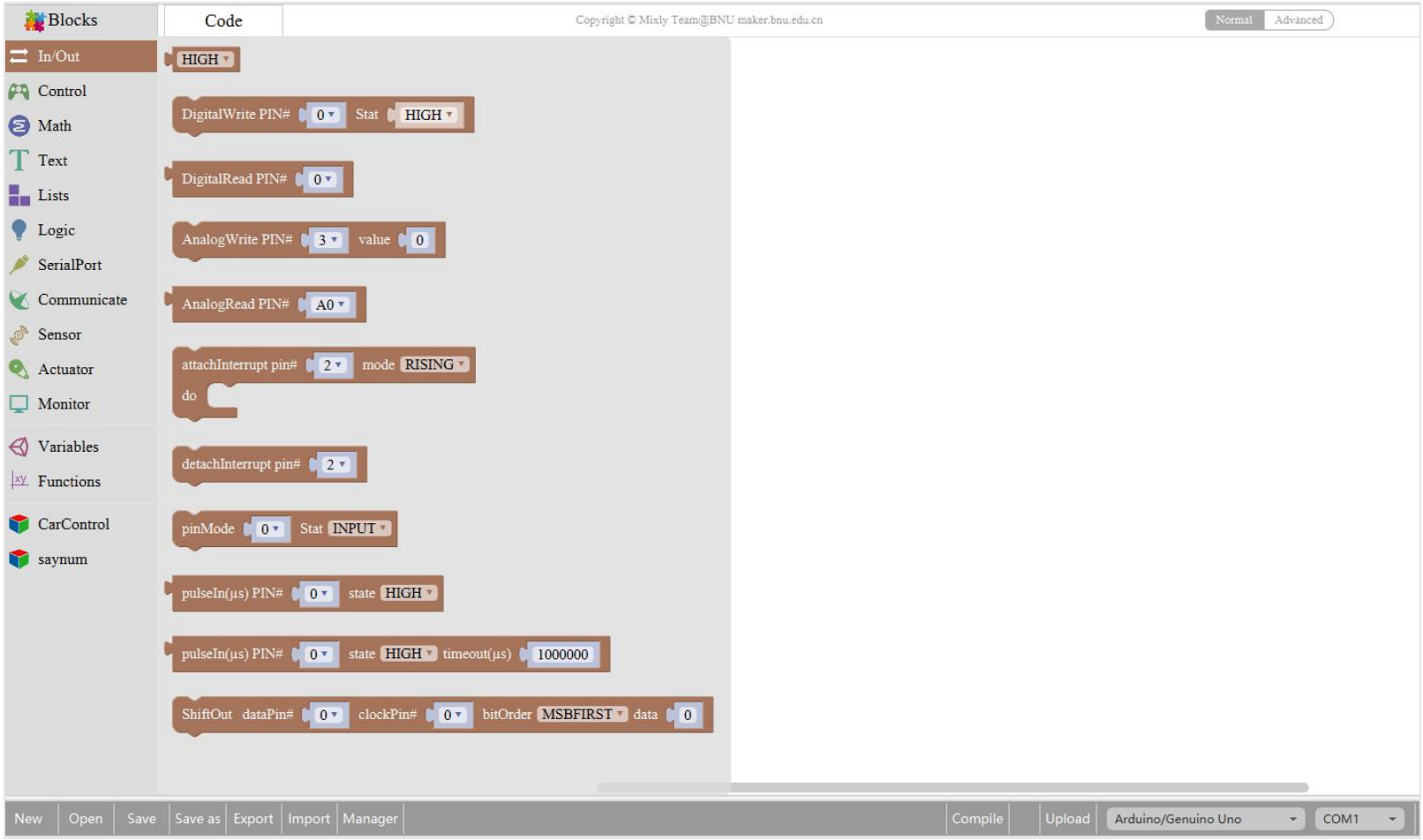


Some common functions:

Through this interface, you can complete the code compile、upload、save and manage. It support four remove methods: drag it left out code window, or drag to Recycle Bin, delete key, or right-click to delete block. It supports four languages: English、Español (Spanish)、中文简体(Chinese Simplified)、中文繁体(Chinese Traditional).



3.2 In/Out Block



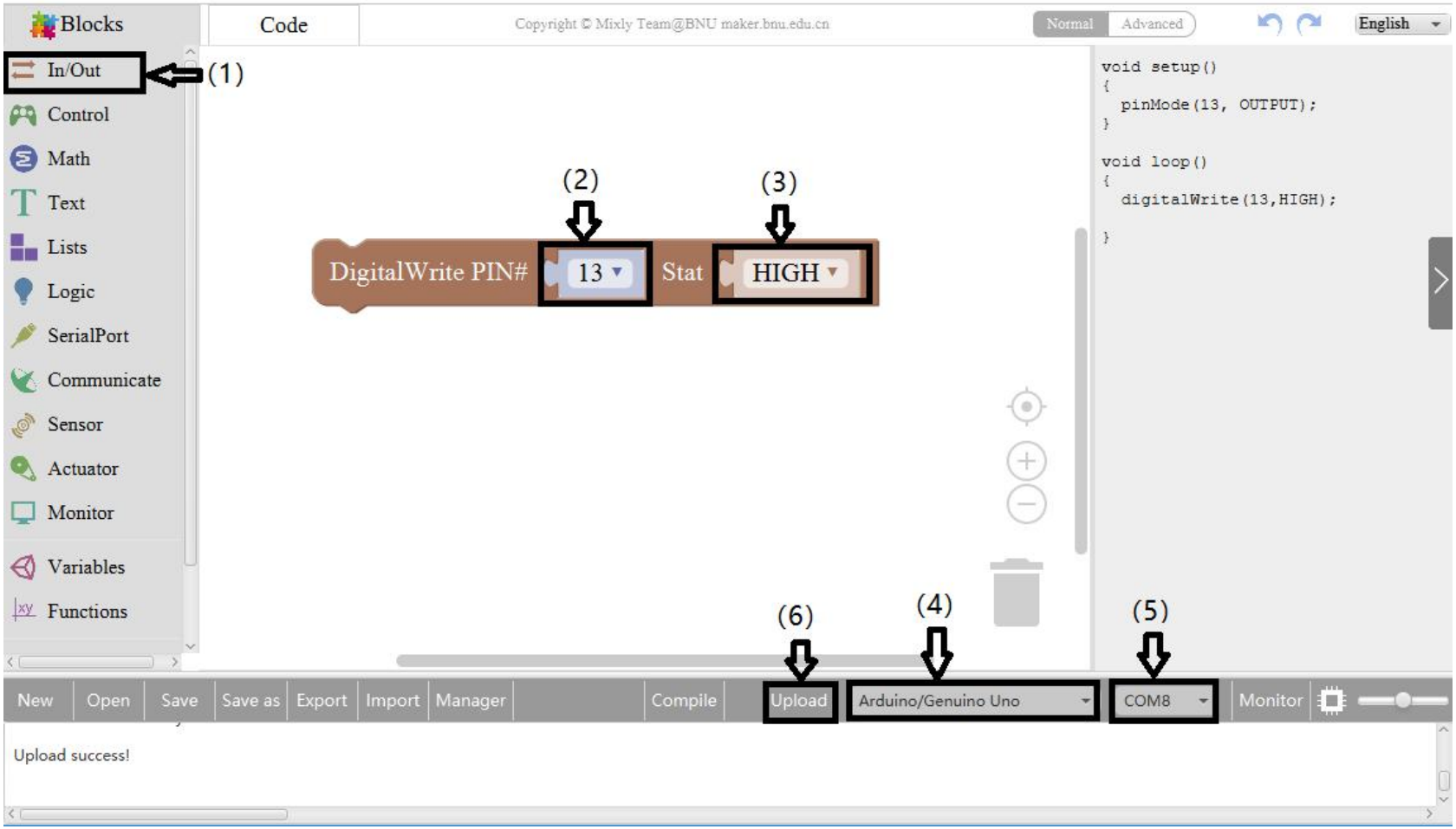
NO.	BLOCK ICON	DEFINITION
1		Returns HIGH or LOW voltage
2		Write digital value to a specific Port. Digital Output: set the HIGH or LOW output for IO pins
3		Returns a digital value of a specific Port. Digital IO Read Pin, generally used to read the HIGH or LOW level detected by Digital sensor
4		Write analog value between 2 and 255 to a specific Port. Analog Output: set the Analog value output by Analog IO pins (0~255).

5		Returns value between 0 and 1023 of a specific Port. Analog IO Read Pin, generally used to read the Analog value detected by Analog sensor.
6		External Interrupts function, with three trigger interrupt modes RISING, FALLING, CHANGE.
7		Detachs interrupt to a specific Port. Turn off the given interrupt function.
8		Set the IO pins as Output or Input state

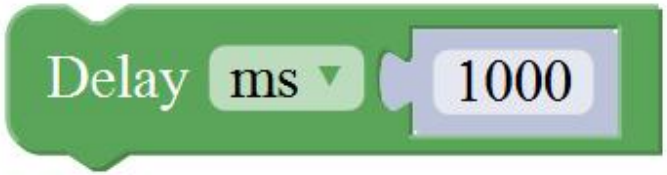
9		Read the continuous time of HIGH or LOW pulse from IO pins (generally used for ultrasonic ranging)
10		Read a pulse (either HIGH or LOW) on a pin within a time set in timeout.
11		Set the ShiftOut data pin, clock pin. Output the data needed from the bitOrder MSBFIRST or LSBFIRST (Most Significant Bit First, or, Least Significant Bit First). Generally used for controlling the 74HC595 CHIP.
12		This is the function interface under Normal mode. If select Advanced mode, the functions will be more.

For example:

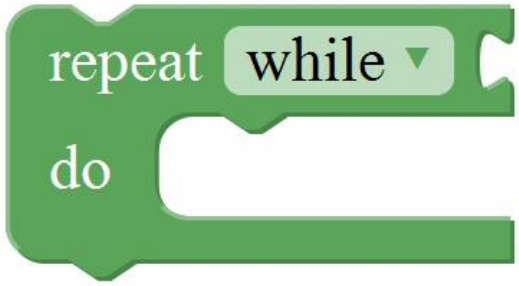
Connect your Arduino Uno board, then follow the steps below to light the Pin13 led on Arduino UNO.



3.3 Control Block

NO.	BLOCK ICON	DEFINITION
1		Initialization (run only once)
2		End the program, means the program will stop running when use this block.
3		Delay function, click to select **ms** or **us** (pause the program for the amount of time (in milliseconds) specified as parameter. There are 1000 milliseconds in a second.)

4		**if_do** function (first evaluate a value be true or false, if a value is true, then do some statement. You can click the blue gear icon to select the **else if** block or **else** block.)
5		**switch** function. You can click the blue gear icon to select the **case** block or **default** block. (used to evaluate several programs then execute the corresponding function matched with program.)
6		Equal to **for** statement.

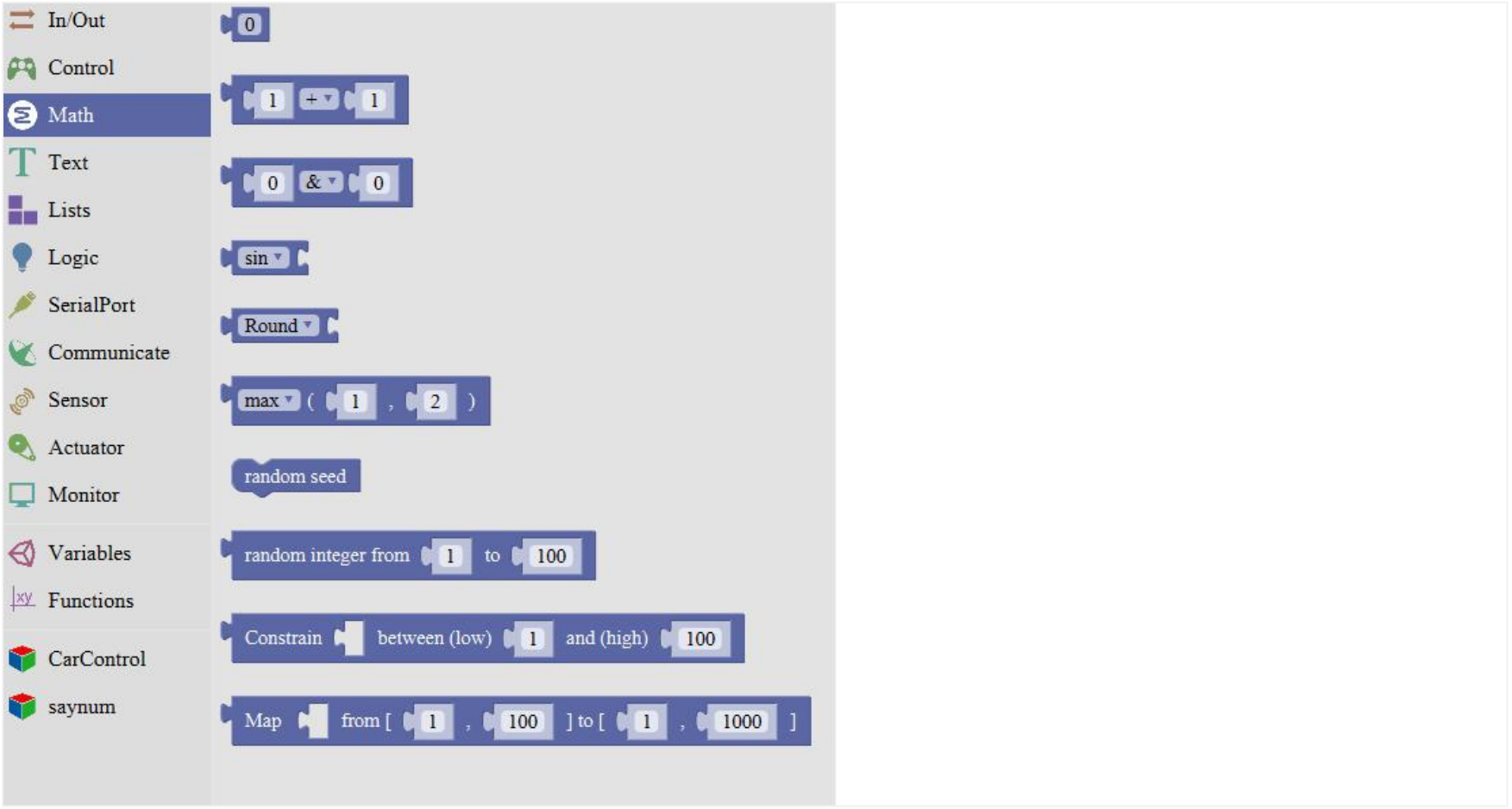
7		A **while** loop statement.
8		**break** function, used to exit from the containing loop.
9		**millis()** function, returns the system running time since the program started. (The unit can be **ms** (milliseconds) or **µs** (microsecond)).

10		Timer interrupt function, that is, set a trigger interrupt for the amount of time (in milliseconds) specified as parameter.
11		Timer interrupt start block
12		Timer interrupt stop block

For example:

Compile and upload the program below to your Arduino board, you should see Pin13 LED on Arduino UNO continue to flash.(with an interval of 1s, equal to 1000ms)

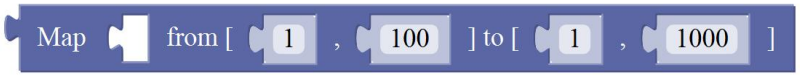
3.4 Math Block



NO.	BLOCK ICON	DEFINITION
1		A number
2		Click to select the Arithmetic Operators: **+(addition); −(subtraction); x (Multiplication); ÷ (division); % (remainder); ^ (bitwise xor)**
3		Click to select the **& (bitwise end); l (bitwise or); << (bitshift left); >> (bitshift right)**

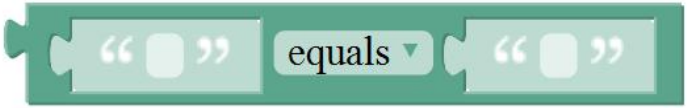
4		Click to select the **sin; cos; tan; asin; acos; atan; ln; log10; e^; 10^; ++ (increment) ; -- (decrement)**
5		Click to select the **Round; Ceil; Floor; abs; sq; sqrt** **Round:** Returns the integer part a number using around. **Ceil:** Returns the integer part a number using ceil. **Floor:** Returns the integer part a number using floor. **abs:** Return the absolute value of a number. **sq:** Return the square of a number. **sqrt:** Return the square root of a number.

6		If select the **max**, returns the larger number; if select the **min**, returns the smaller number.
7		Initialize the random seed
8		Return a random integer between the two specified limits, inclusive.
9		Constrain a number to be between the specified limits (inclusive). (generally used to constrain an analog value read from sensor)

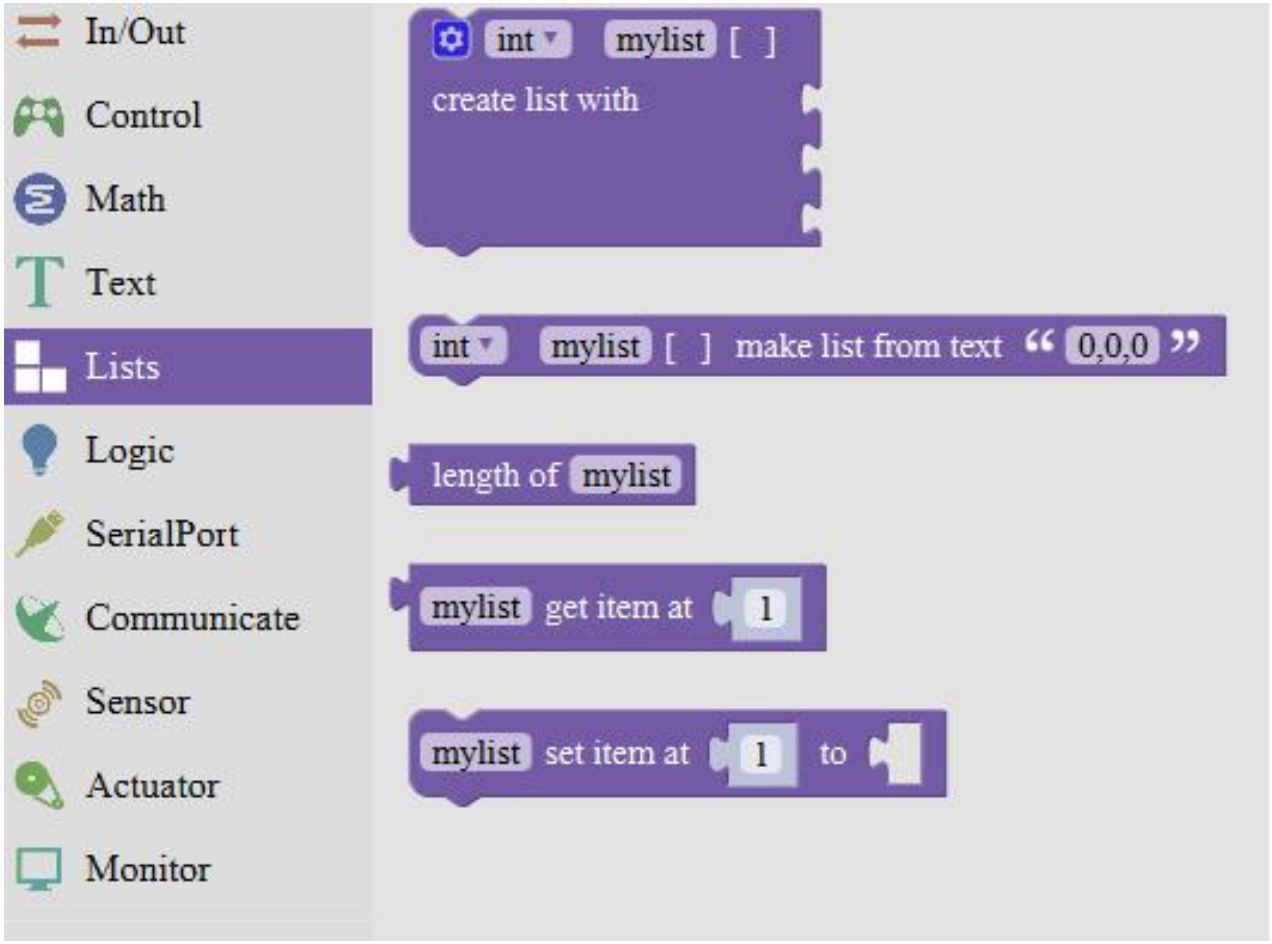
10		Map a number from the first interval to the second interval. (For instance, potentiometer-controlled servo, map the range of potentiometer (0, 1023) to the angle of servo (0, 180)).

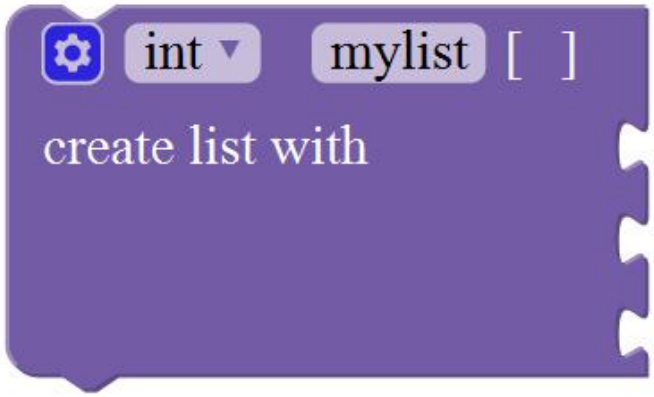
3.5 Text Block

NO.	BLOCK ICON	DEFINITION
1		character string: a letter, word, or line of text.
2		A character
3		Creates a piece of text by joining together two piece of text. (Here Hello join Mixly equals HelloMixly)
4		Converts a string into an integer or an float.
5		Returns the char corresponding to an ASCII code (Decimal number 97 corresponding to a)

6		Returns the ASCII code corresponding to a char.
7		Converts a number into a string.
8		Calculates the length of a string
9		Output the char of a string (the char at 0 of hello is h)
10		The first string equals or startsWith or endsWith the second string, returns 1, otherwise returns 0. (if equals, both strings are abc, returns 1.)
11		Returns a decimal value of the first string subtracts the second string.

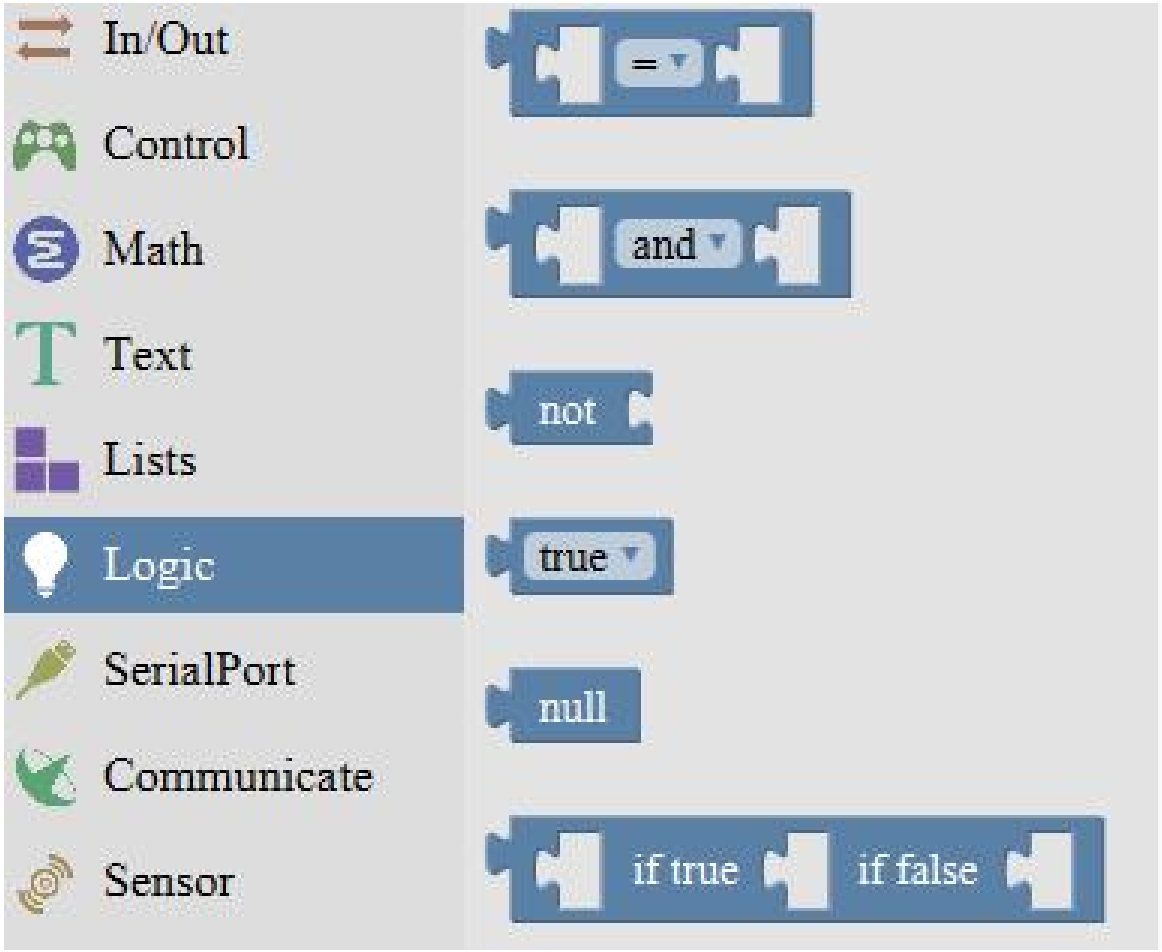
3.6 List Block

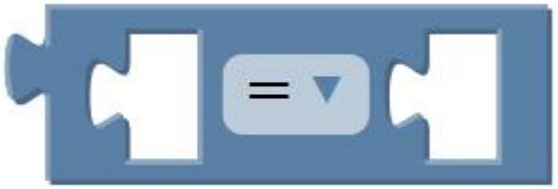


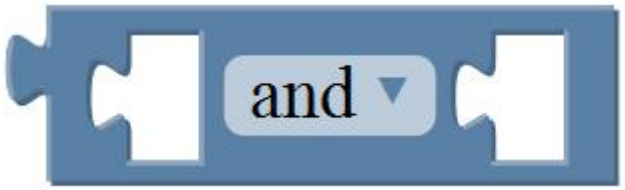
NO.	BLOCK ICON	DEFINITION
1		Create a list with any number of items
2		Creats a list from a text. (int mylist [ ]={0,0,0};)
3		Returns the length of a list

4		Returns the value of at the specified position in a list.
5		Sets the value of at the specified position in a list. Set the first item in mylist to another item.

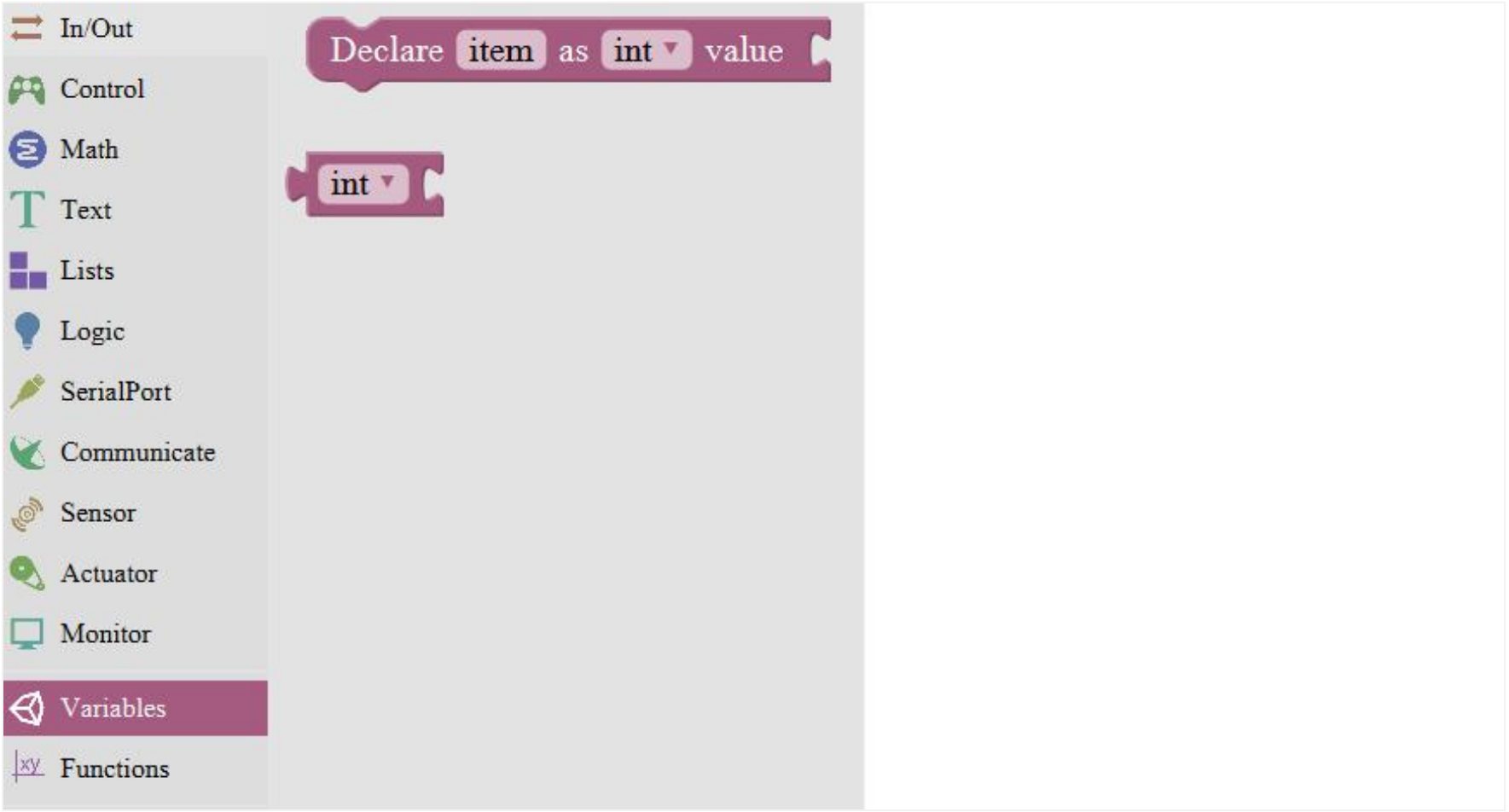
3.7 Logic Block

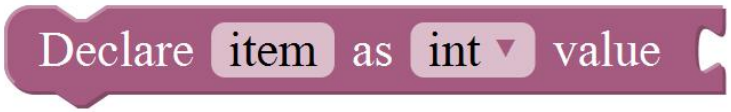
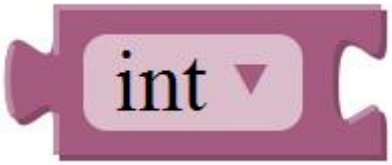


NO.	BLOCK ICON	DEFINITION
1		**logic comparision** **=**: Return true if both inputs equal each other. **≠** : Return true if both inputs are not equal to each other. **<**: Return true if the first input is smaller than the second input. **≤** : Return true if the first input is smaller than or equal to the second input. **>**: Return true if the first input is greater than the second input. **≥ :** Return true if the first input is greater than or equal to the second input.

2		**and:** Return true if both inputs are true; **or:** Return true if at least one of the inputs is true
3		Returns true if the input is false. Returns false if the input is true.
4		Returns either true or false.
5		Returns null
6		If the first number is true, the second number is returned, otherwise the third number.

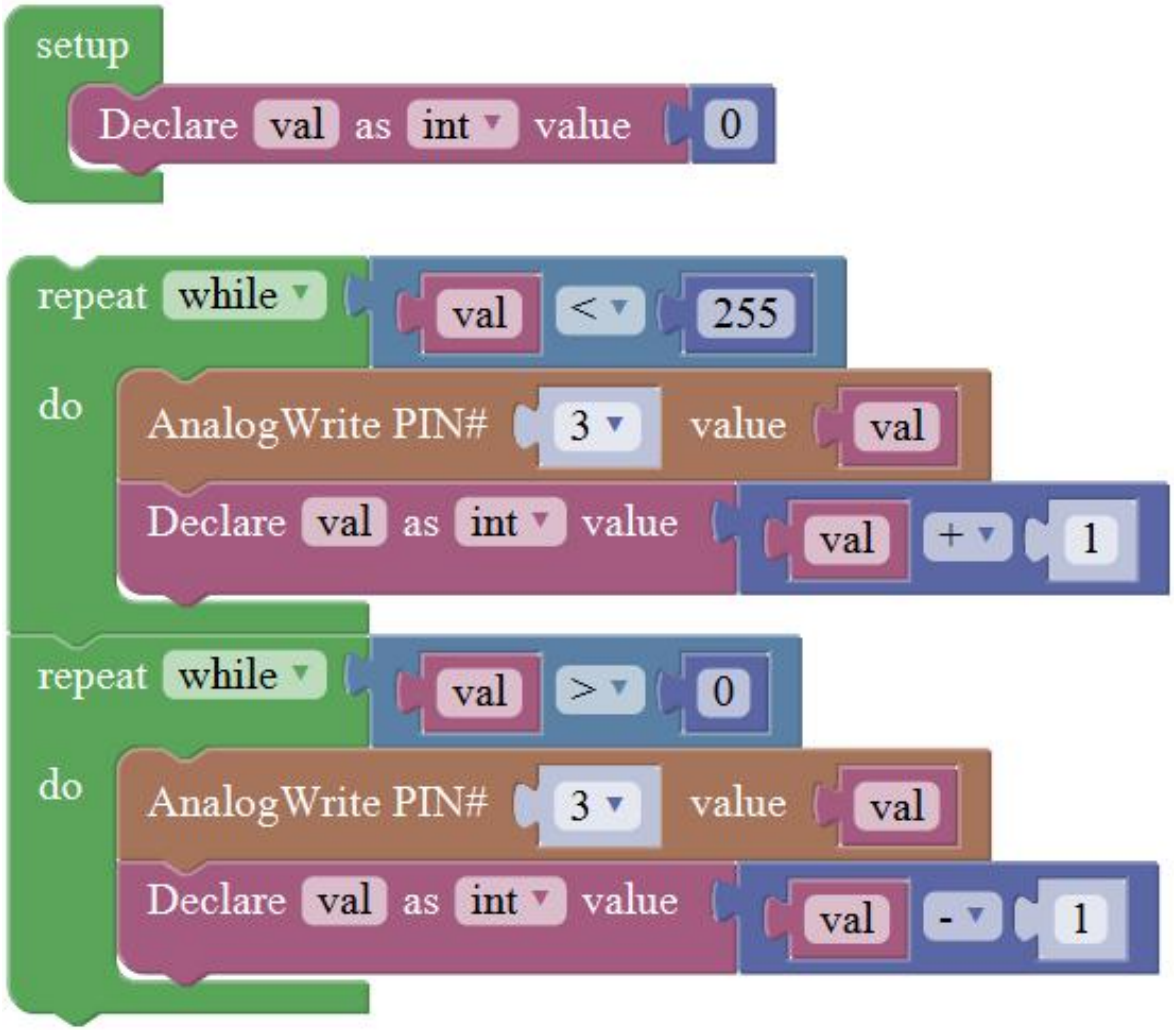
3.8 Variable Block



NO.	BLOCK ICON	DEFINITION
1		Declare and initialize a variable. Click to select **int**, **long**, **float**, **boolean**, **byte**, **char**, **string**
2		Define the data types

For example: LED breath

You need an Arduino Uno and one LED module. Connect the control pin of LED module to Pin 3 of Uno board (or other pins with “~”, that is, those pins can output PWM signal). LED will gradually light then gradually dim, repeatedly.


setup
Declare val as int value 0
repeat while val < 255
do
AnalogWrite PIN# 3 value val
Declare val as int value val + 1
repeat while val > 0
do
AnalogWrite PIN# 3 value val
Declare val as int value val - 1

3.9 SerialPort Block

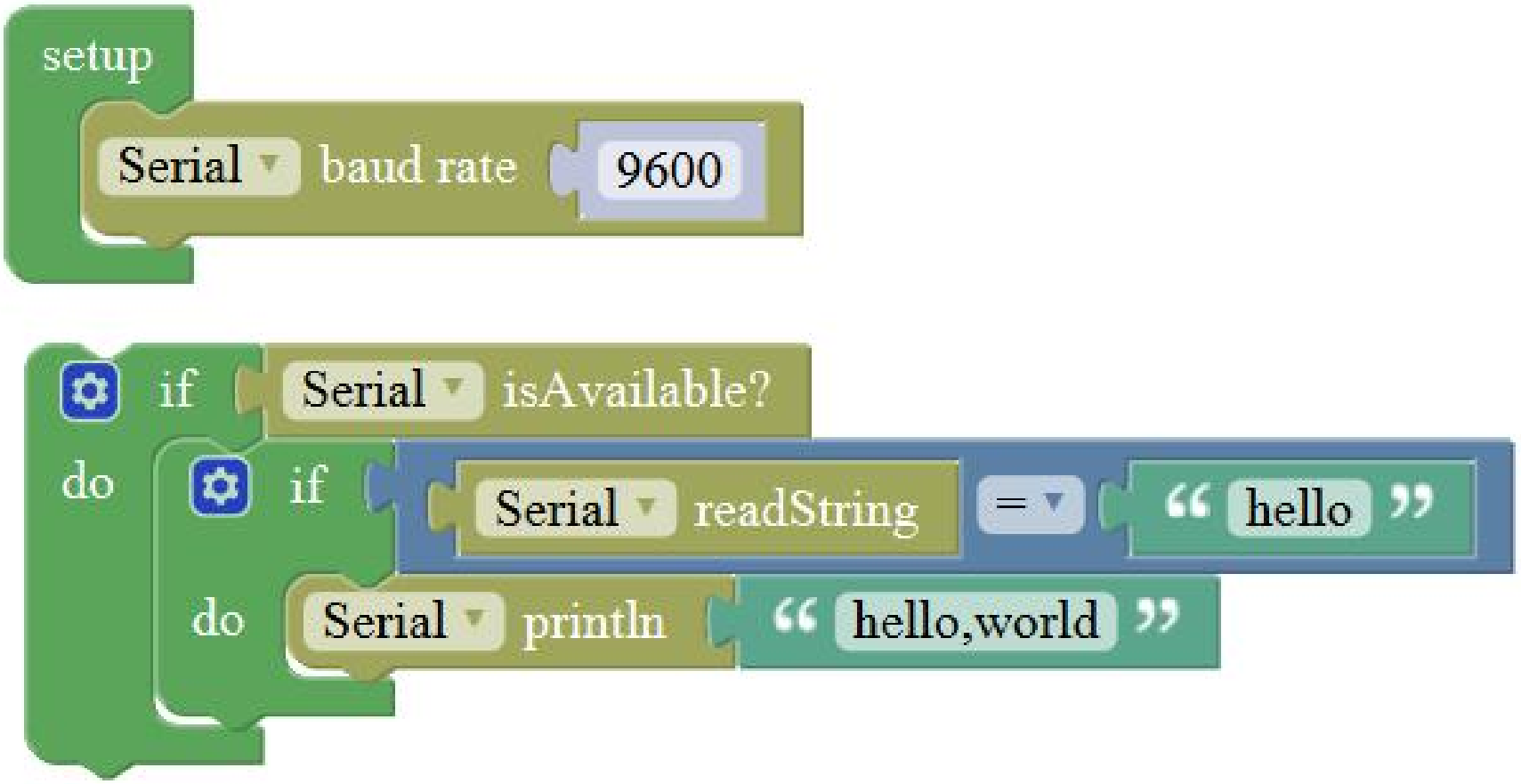
NO.	BLOCK ICON	DEFINITION
1		Set the serial buad rate to 9600
2		Write the specified number, text or other value.
3		Print the specified number, text or other value on monitor.
4		Print the specified number, text or other value on newline of monitor.

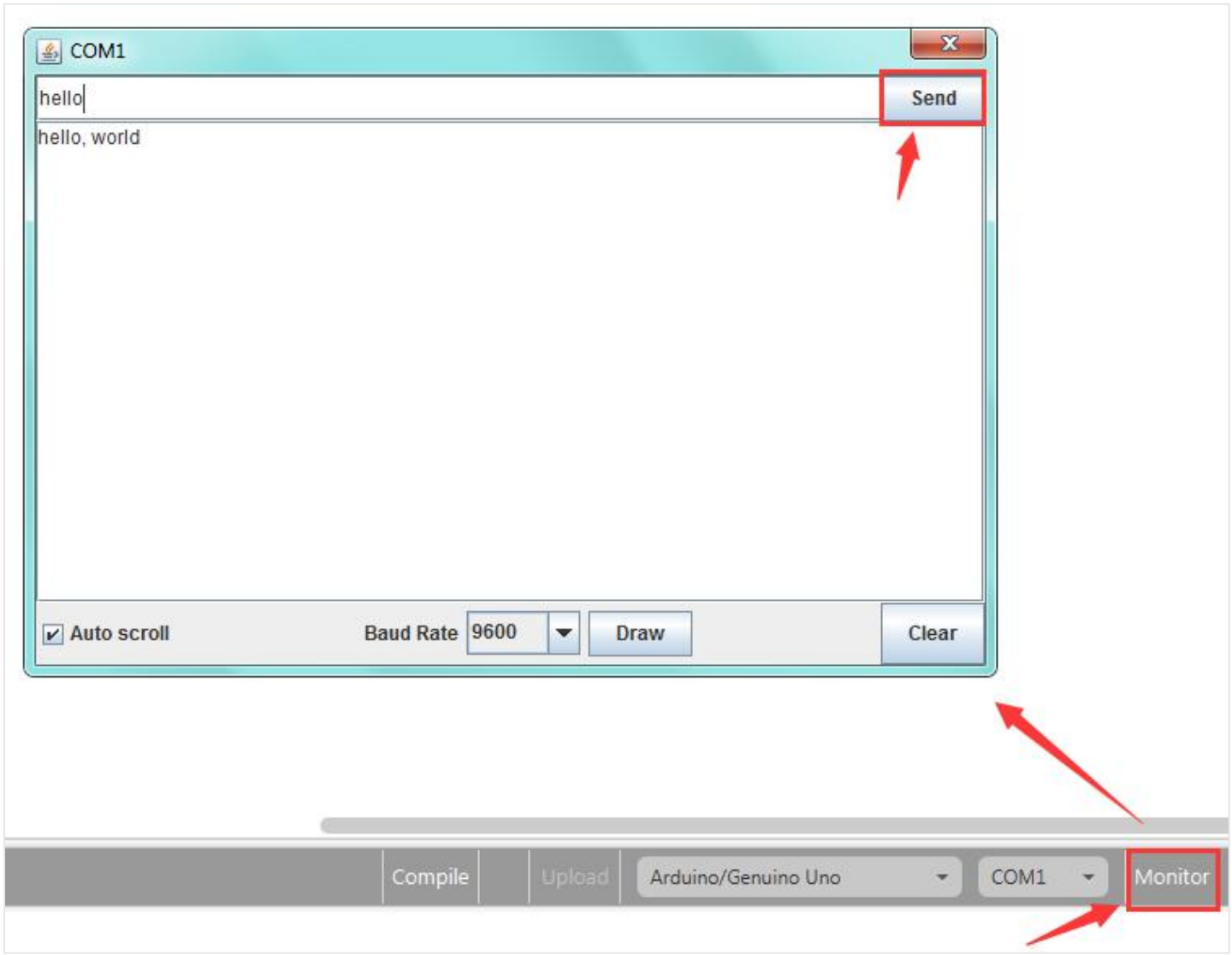
5		Print the specified number in hexademical format on newline of monitor.
6		If the serial port is available, it returns true, otherwise returns false. (generally used in Bluetooth communication)
7		Returns a string in serial port
8		A string read from serial port to a string variable, pause until read the specified character.
9		Read the serial data by byte (generally used to read the value sent from Bluetooth) (delete the data has been read)

10		Wait for the output data completed
11		Set the software serial port (call this function if need to use several serial ports)
12		Event function trigger by serial port data, that is, serial port is ready to call this function. (equal to an interrupt function)

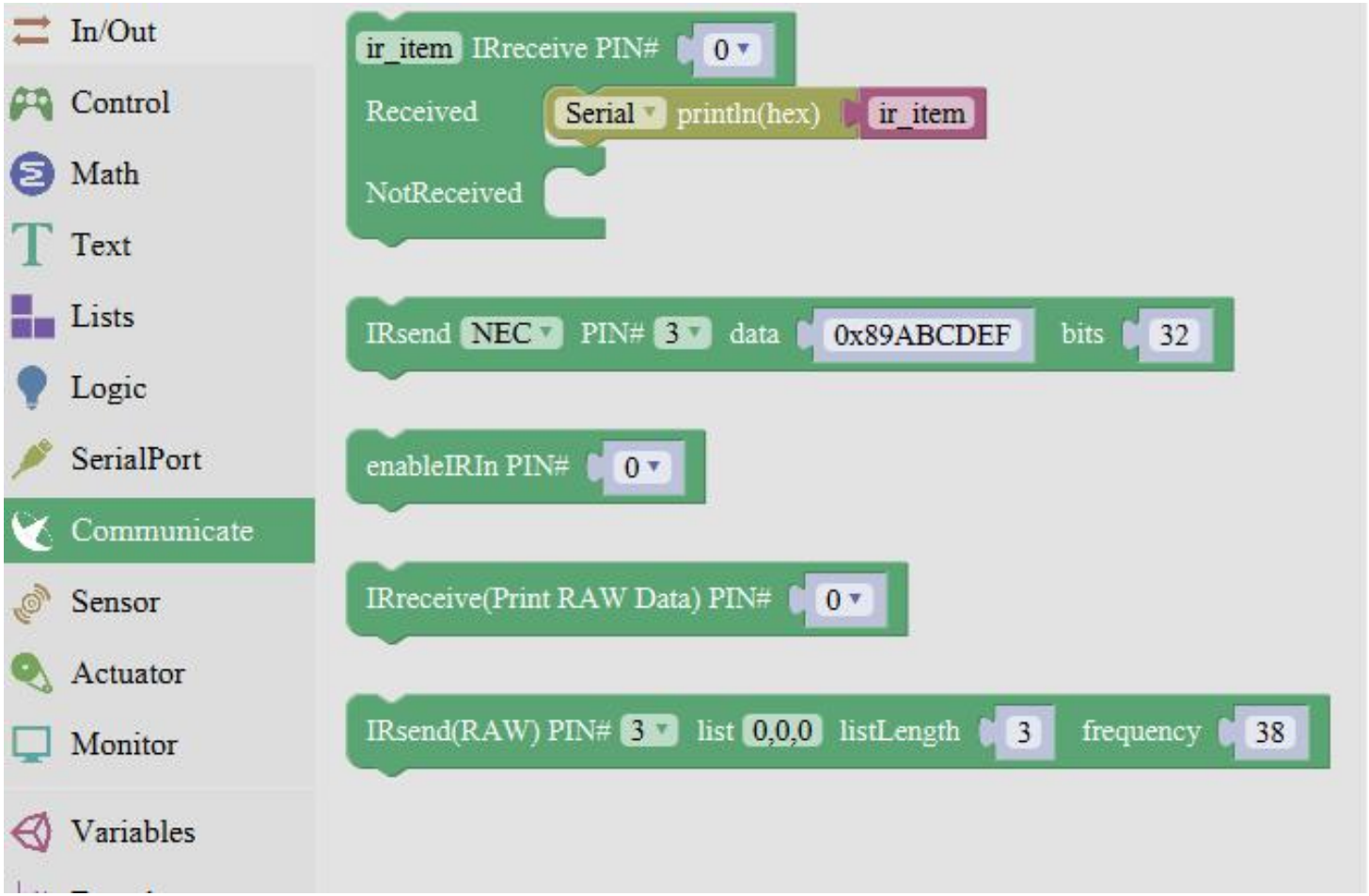
For example: serial communication

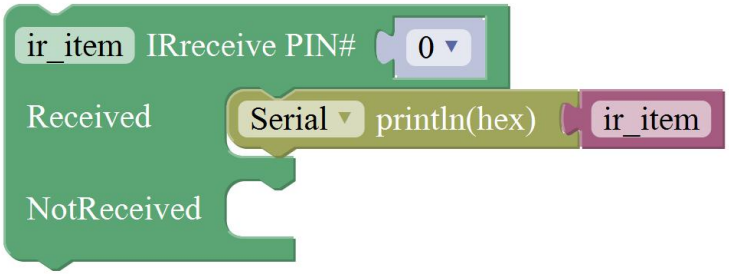
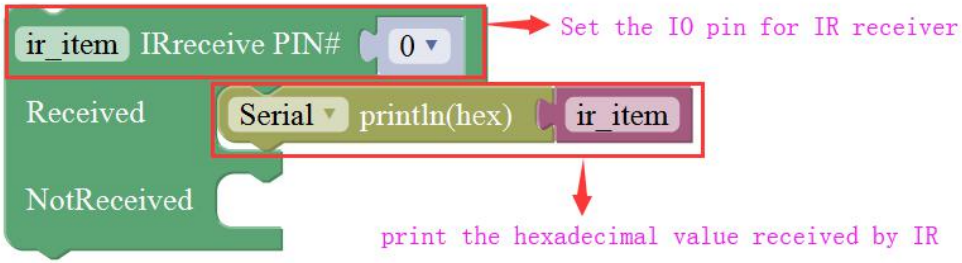
Done uploading the code, open the Arduino monitor, then enter a “hello” on the top bar, and click Send, it will print out “hello,world”.


setup
Serial
baud rate
9600
if
Serial
isAvailable?
do
if
Serial
readString
=
“
hello
”
do
Serial
println
“
hello,world
”


COM1
hello
Send
hello, world
Auto scroll
Baud Rate
9600
Draw
Clear
Compile
Upload
Arduino/Genuino Uno
COM1
Monitor

3.10 Communicate Block



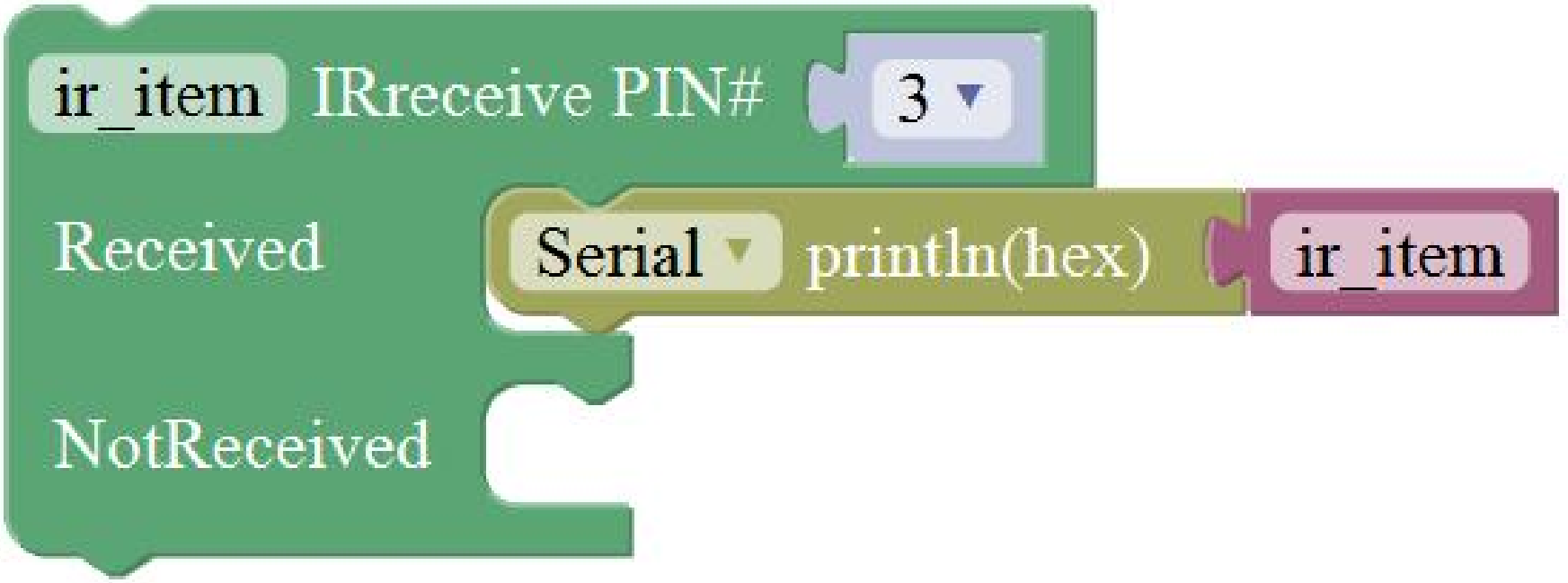
NO.	BLOCK ICON	DEFINITION
1		Do something when receiving infrared signals. 
2		Sends infrared signals of the specified types. IR transmitter sends the data, here use the libraries, only PIN3 port.

3		Enable IR decoding
4		Print the Infrared signal in RAW types when receiving it.
5		Sends RAW infrared signals (set the pin number, list, length of list and IR frequency)

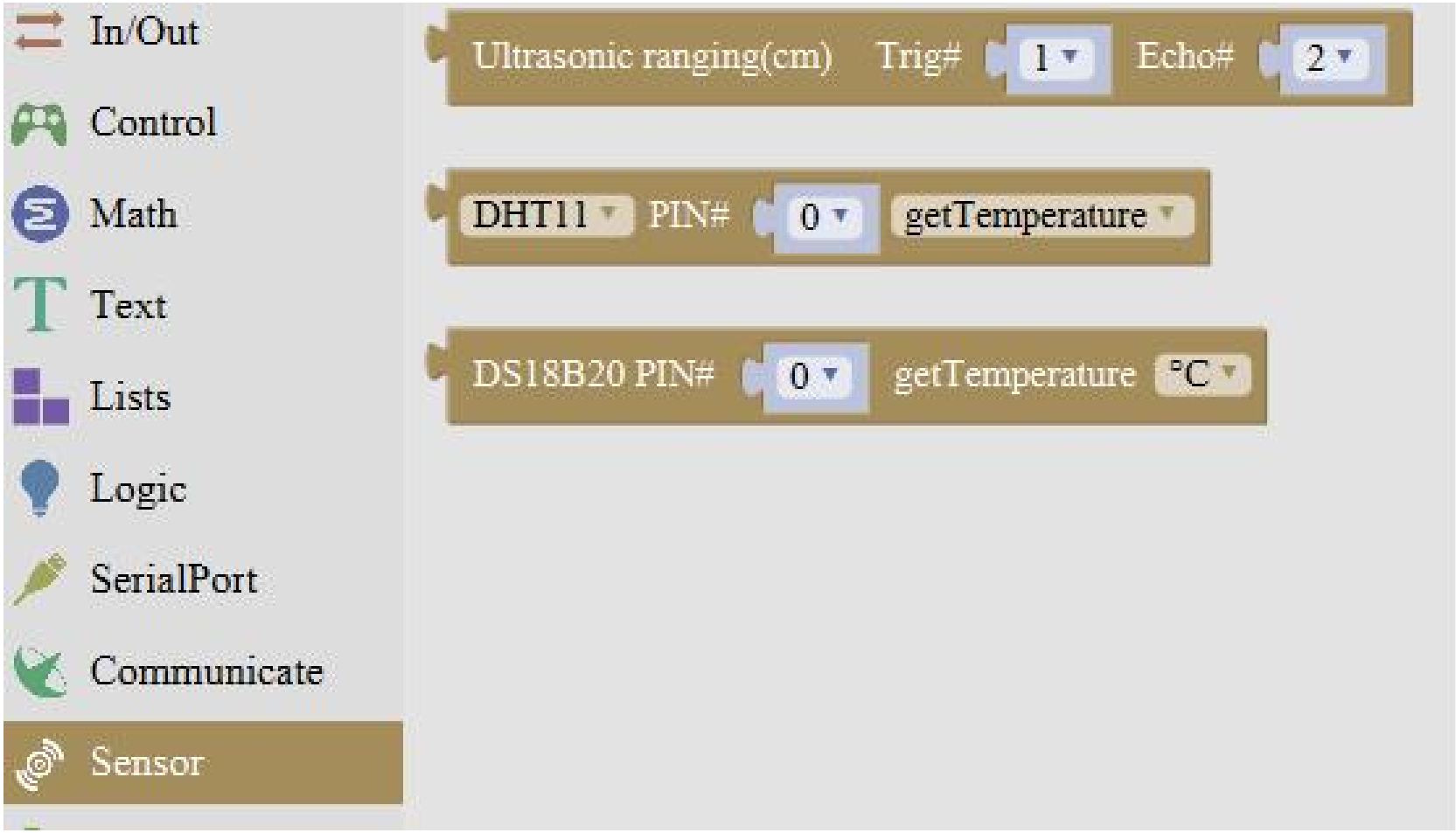
For example:

You need an Arduino Uno board, an IR receiver module and an IR remote control.

Connect the signal pin of IR receiver to Digital pin 3 of Uno board, then upload the code and open the monitor. If send a signal to an IR receiver module using an IR remote control, you should see the monitor show the corresponding signal data.


ir_item IRreceive PIN# 3
Received
Serial println(hex) ir_item
NotReceived

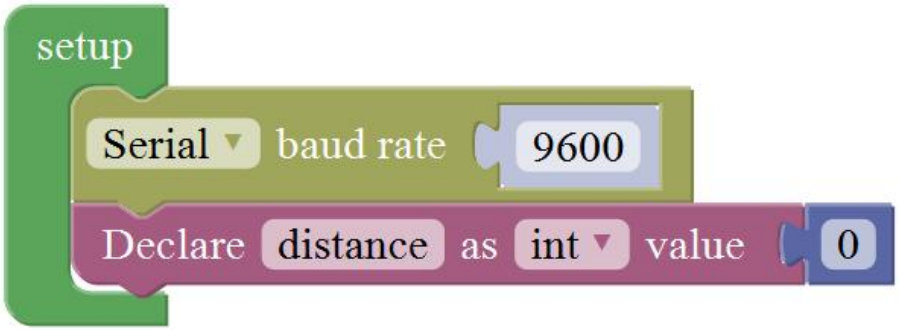
3.11 Sensor Block

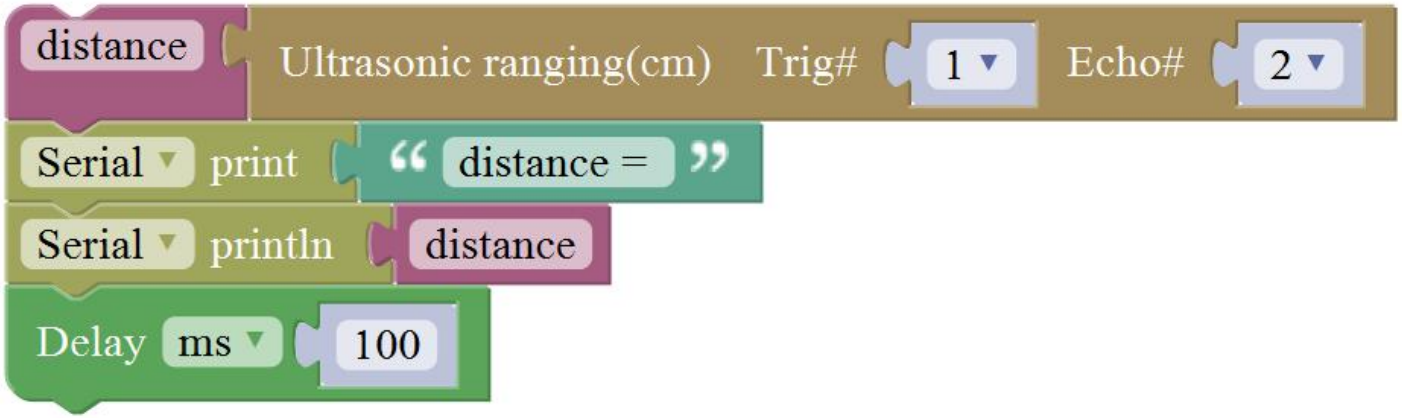


NO.	BLOCK ICON	DEFINITION
1		Set the Trig and Echo pin of ultrasonic sensor. Returns the distance of ultrasonic sensor measured. (**unit:** cm)
2		Set the control pin of DHT11 temperature and humidity sensor. Returns the temperature or humidity of DHT 11 sensor measured.
3		Set the pin of digital temperature sensor DS18B20. Returns the temperature value of DS18B20 sensor measured.

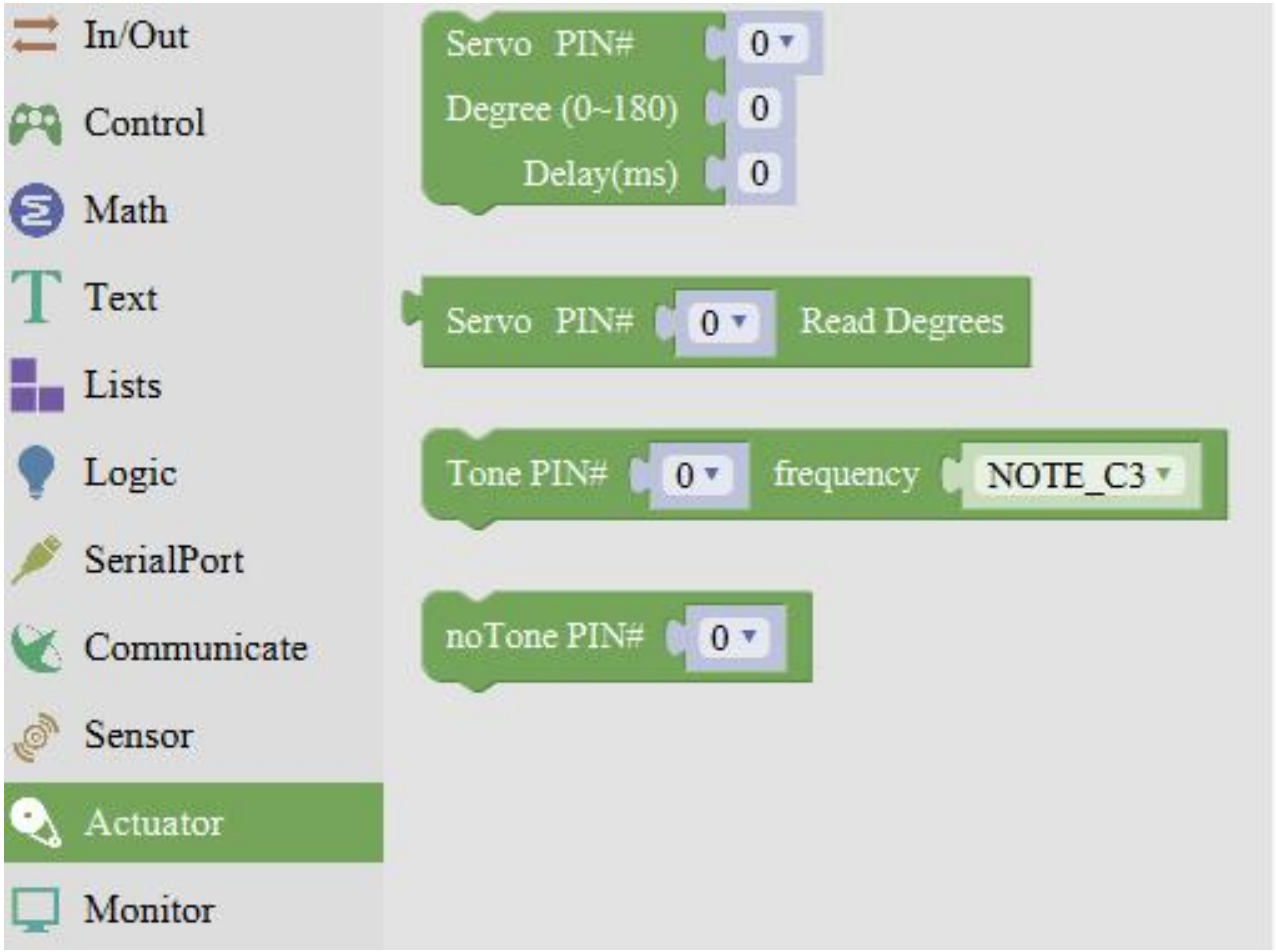
For example:　ultrasonic ranging

Connect the Trig pin of ultrasonic sensor to Digital 1 of Uno, Echo pin to D2, then upload the code and open the monitor, you should see the distance value, updating once per 100ms.





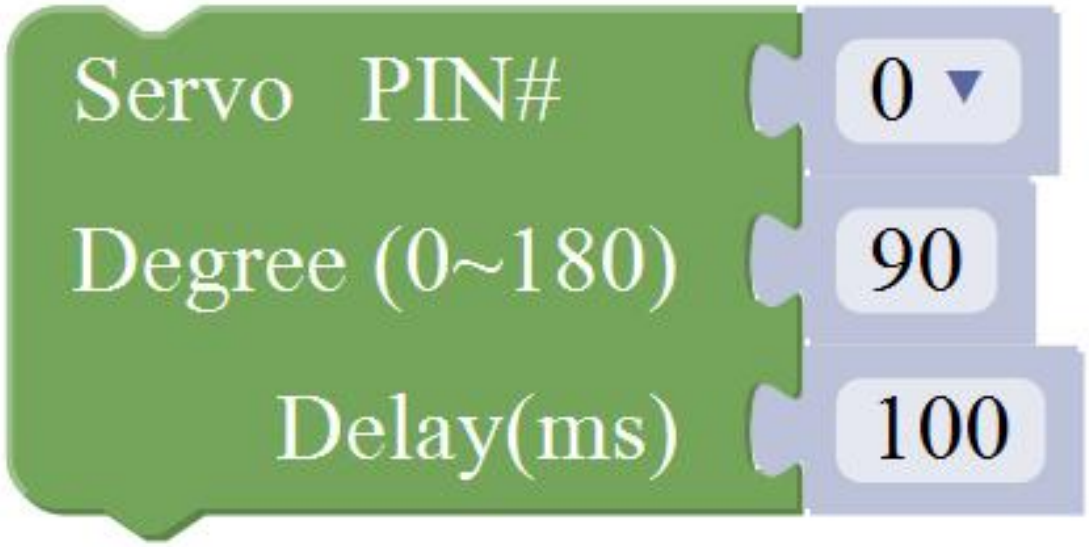
3.12 Actuator Block



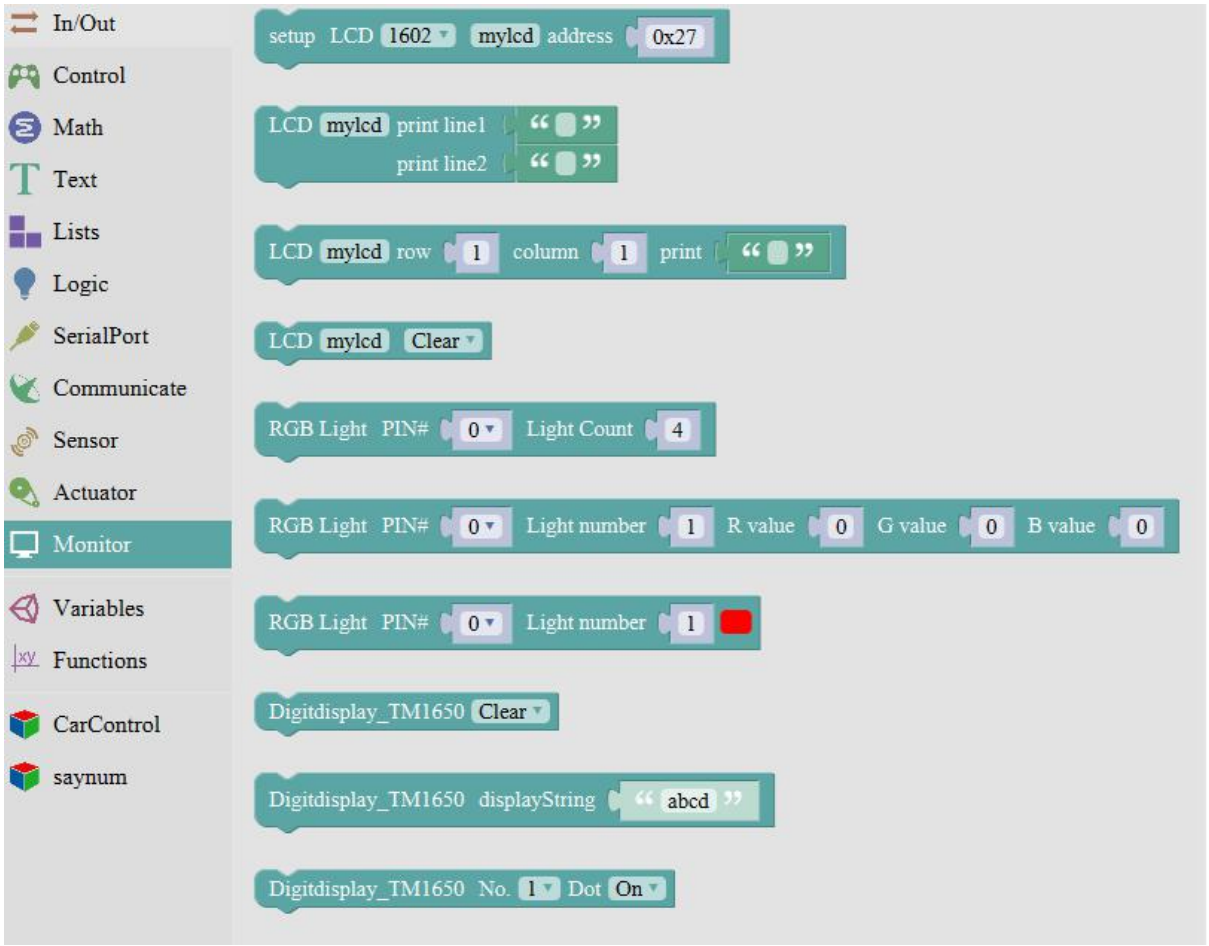
NO.	BLOCK ICON	DEFINITION
1		Sets the servo pin; Moves between 0-180 degree; Delay time for servo to rotate.
2		Returns that degree with the last servo move. Read the degree of servo connected to IO pin set
3		Set the pin and specified frequency for buzzer to play sound.
4		Stop playing sound

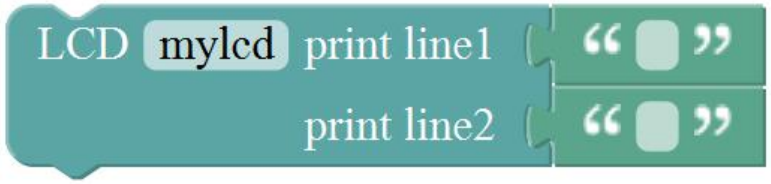
For example:

Connect the signal end of servo to Digital 0 of Uno, then upload the code below, servo will rotate 90 degrees. Note: Delay 100ms is the time required for servo to move.



3.13 Monitor Block

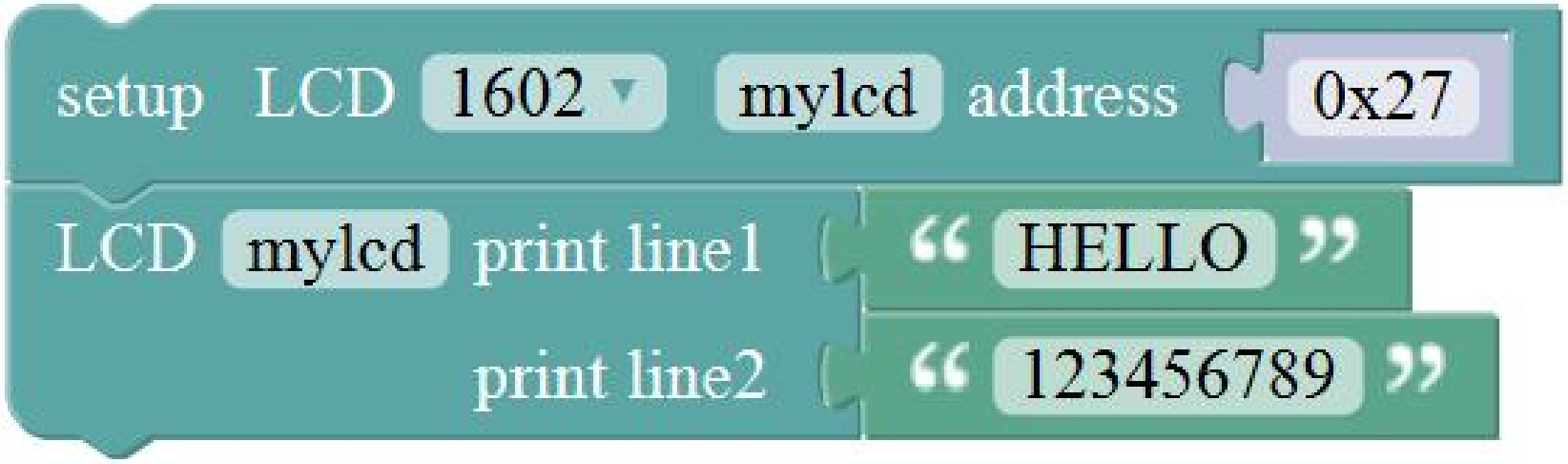


NO.	BLOCK ICON	DEFINITION
1		Set the IIC LCD1602 address
2		Input the value on LCD line 1 and line 2 from left to right.
3		Set the row and column of LCD to print the char
4		Clear the LCD screen
5		Set the control pin and the number of RGB light.

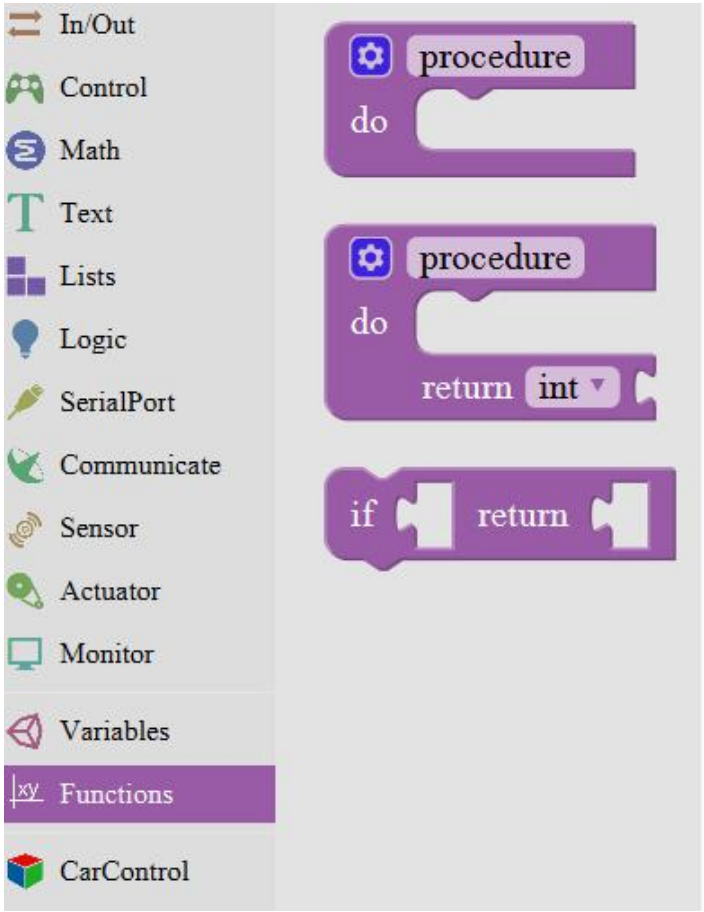
6		Set the RGB light pin, light number and brightness
7		Set the control pin, light number and color. (click to select the color)
8		Clear the data, namely turn off digital display
9		Four-digit display, displaying abcd.
10		Turn on or off the digitdisplay (here turn on the first digitdisplay)

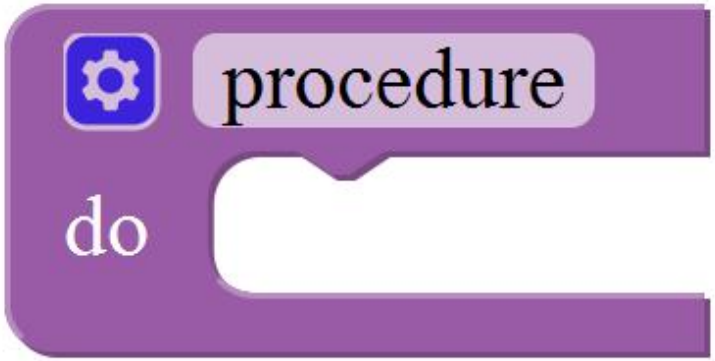
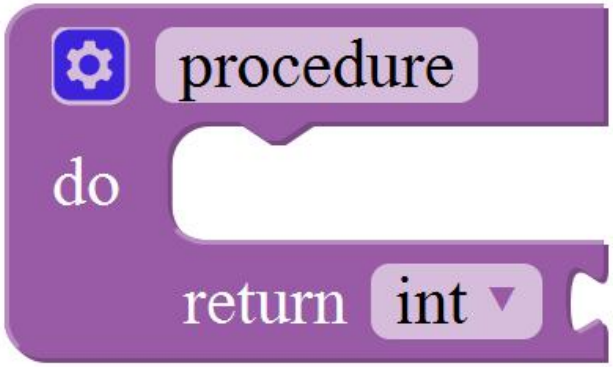
For example:

Separately connect the SDA (A4) and SCL (A5) of Arduino Uno to SDA and SCL pins of IIC LCD1602, then set the address of your LCD1602 screen, the LCD address we used here is 0x27. Then upload the code, LCD screen has two lines, you should see the line 1 print HELLO, and line 2 print 123456789.



3.14 Functions Block

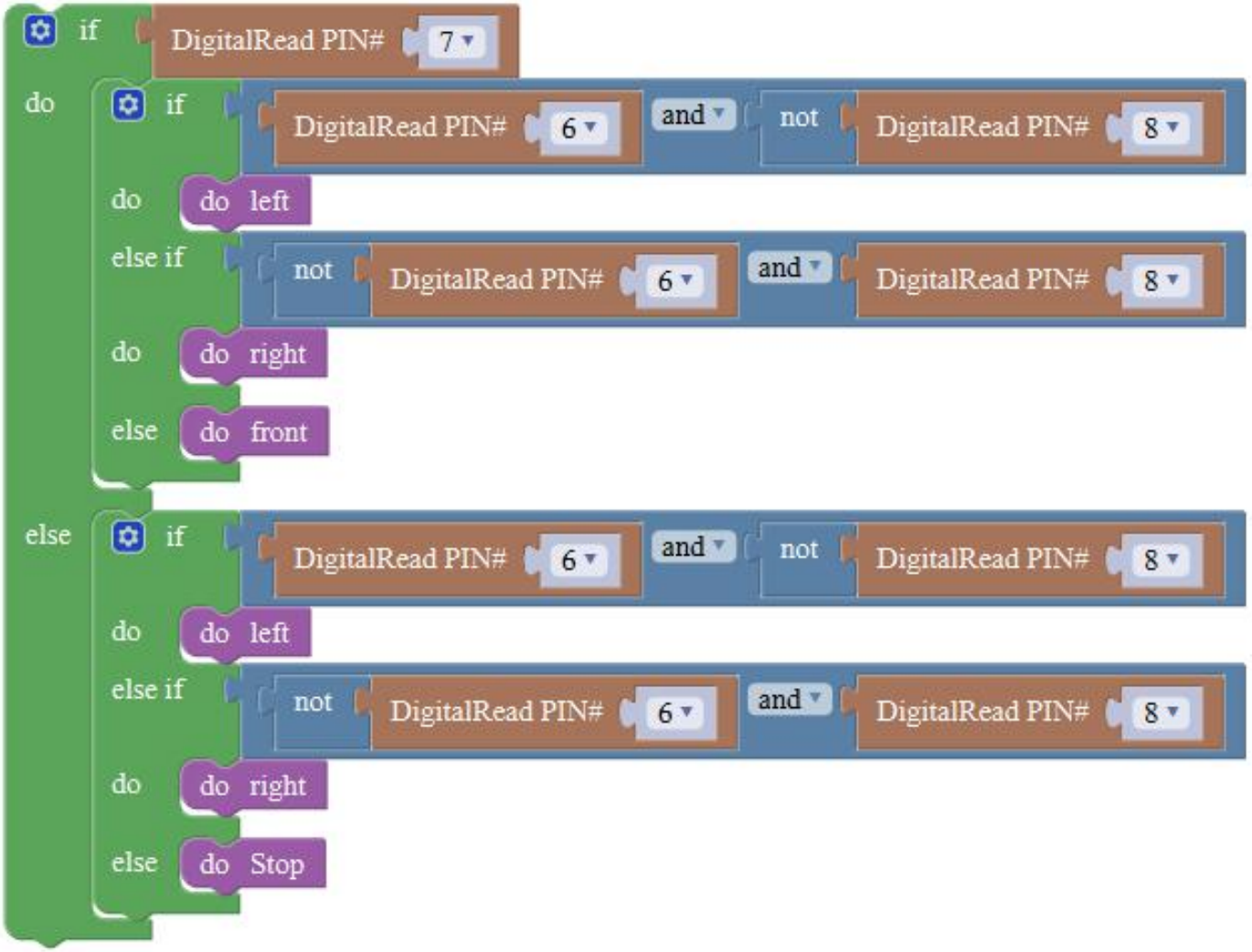


NO.	BLOCK ICON	DEFINITION
1		Creates a function with no output. Click the blue icon to set the procedure parameter. (no return value)
2		Creates a function with an output. Click the blue icon to set the procedure parameter. (with return value and can set the data types)

3		If a value is true, then return a second value.

For example:

Below is an example code for line tracking car. We use three tracking modules (left to D6, middle to D7, right to D8). of course you need a tracking car to test it. First edit the forward, backward, turn left, turn right and stop into functions block. Then compile and upload the code below.


if
DigitalRead PIN# 7
do
if
DigitalRead PIN# 6
and
not
DigitalRead PIN# 8
do
do left
else if
not
DigitalRead PIN# 6
and
DigitalRead PIN# 8
do
do right
else
do front
else
if
DigitalRead PIN# 6
and
not
DigitalRead PIN# 8
do
do left
else if
not
DigitalRead PIN# 6
and
DigitalRead PIN# 8
do
do right
else
do Stop


front
do
DigitalWrite PIN# 4 Stat LOW
DigitalWrite PIN# 2 Stat LOW
AnalogWrite PIN# 9 value 200
AnalogWrite PIN# 5 value 200
left
do
DigitalWrite PIN# 4 Stat HIGH
DigitalWrite PIN# 2 Stat LOW
AnalogWrite PIN# 9 value 200
AnalogWrite PIN# 5 value 200
baxk
do
DigitalWrite PIN# 4 Stat HIGH
DigitalWrite PIN# 2 Stat HIGH
AnalogWrite PIN# 9 value 200
AnalogWrite PIN# 5 value 200
right
do
DigitalWrite PIN# 4 Stat LOW
DigitalWrite PIN# 2 Stat HIGH
AnalogWrite PIN# 9 value 200
AnalogWrite PIN# 5 value 200
Stop
do
DigitalWrite PIN# 4 Stat HIGH
DigitalWrite PIN# 2 Stat HIGH
AnalogWrite PIN# 9 value 0
AnalogWrite PIN# 5 value 0

4. Resources

Download the Mixly software:

https://drive.google.com/open?id=1oQxF-AZ0Aw6OQhu_8NSvwo3L2OP0Z6cU

Download the Example Code:

https://drive.google.com/open?id=1Fjd3SHHkg_-0IdB6GPX2uuTv3aRGMCSd